

Лабораторная работа №10

Препроцессорные директивы

Контрольные вопросы:

- 1) Что такое компиляция программ и зачем она нужна?
- 2) Какие типы компиляции есть в языке C?
- 3) Что такое этап препроцессинга и зачем он нужен?
- 4) Что такое этап компиляции и зачем он нужен?
- 5) Что такое этап ассемблирования и зачем он нужен?
- 6) Что такое этап компоновки и зачем он нужен?
- 7) Что такое объектный файл и для чего он нужен?
- 8) Что такое таблица символов и для чего она нужна?
- 9) Для чего нужна директива `#include` и каков ее формат?
- 10) Для чего нужна директива `#define` и каков ее формат?
- 11) Что такое условная компиляция?

Задание

Модифицировать программу из лабораторной работы №8 таким образом, чтобы в ней использовались препроцессорные директивы для следующего:

- 1) Чтение\запись элементов одномерных массивов (которые в 0-ом элементе хранят длину массива) идёт только через макрос `ELEM(array,i)` причём в макросе индекс `i` считается с нуля.
- 2) Чтение\запись значения длины одномерных массивов идёт только через макрос `LEN(array)`
- 3) Тип элементов массивов задаётся через макроопределение, например можно объявить массивы с элементами типа `short`, `int` и `long long`
- 4) Использовать условную компиляцию и в зависимости от объявленных `define`-ов изменить минимум два типа поведения функций обработки одномерных массивов на выбор студента. Примеры изменения типов поведения функций: найти не первый, а последний элемент; не с конца, а с начала; не минимум, а максимум; сдвиг не влево, а вправо и т.д.
- 5) Вынести все функции по работе с одномерными массивами в отдельный файл `funcs.inc` и подключить его к исходному файлу через `#include`.